JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 63, 396–411 (1978)

A Curvilinear Optimization Method Based upon Iterative Estimation of the Eigensystem of the Hessian Matrix

C. A. BOTSARIS

National Research Institute for Mathematical Sciences of the Council for Scientific and Industrial Research, P.O. Box 395, Pretoria, South Africa

Submitted by E. Stanley Lee

An algorithm was recently presented that minimizes a nonlinear function in several variables using a Newton-type curvilinear search path. In order to determine this curvilinear search path the eigenvalue problem of the Hessian matrix of the objective function has to be solved at each iteration of the algorithm. In this paper an iterative procedure requiring gradient information only is developed for the approximation of the eigensystem of the Hessian matrix. It is shown that for a quadratic function the approximated eigenvalues and eigenvectors tend rapidly to the actual eigenvalues and eigenvectors of its Hessian matrix. The numerical tests indicate that the resulting algorithm is very fast and stable. Moreover, the fact that some approximations to the eigenvectors of the Hessian matrix are available is used to get past saddle points and accelerate the rate of convergence on flat functions.

1. INTRODUCTION

In [1] a new algorithm for function minimization was presented. This algorithm generates a sequence of approximate minimizers via the iteration formula

$$x^{k+1} = x^k + [\exp(-t(x^k)\, H(x^k)) - I]\, H^{-1}(x^k)\, g(x^k), \tag{1.1}$$

where $g(x)$ and $H(x)$ are the gradient vector and the Hessian matrix of the objective function $f(x)$, respectively, and $t(x^k)$ is the value of a curve parameter t so chosen that a function decrease will result when moving from x^k to x^{k+1} along the curvilinear path. Although the resulting algorithm minimizes a quadratic function in one step and rapidly minimizes general functions, it has two disadvantages:

(i) the second partial derivatives of the objective function, i.e., the Hessian matrix, must be computed analytically; and

(ii) the eigenvalue problem of $H(x)$ must be solved at each iteration.

At this point we recall that the computation of second partial derivatives may be too difficult for a highly nonlinear objective function and that the solution

0022-247X/78/0632–0396$02.00/0

of the eigenvalue problem of the Hessian matrix is time-consuming, especially for minimization problems of high dimensionality.

Although in [2, 3] an iterative procedure was developed which approximates the Hessian matrix using gradient information only, an eigenvalue problem still had to be solved.

In this paper we present an iterative scheme for the approximation of the eigenvalues and eigenvectors of the Hessian matrix themselves. The resulting algorithm does not possess the property of quadratic termination, but the approximated eigenvalues and eigenvectors tend, under certain conditions, to those of the actual Hessian matrix when the objective function is quadratic. In the approximation scheme derived in this paper gradient information only is used, and the method is a generalization of the inverse iteration method for solving the algebraic eigenvalue problem [5].

2. Preliminaries

Let $f: R^n \to R^1$ be a nonlinear function having a unique minimum at x^*, and let us consider the initial-value system of differential equations

$$\dot{x}(t) = -g(x), \qquad x(0) = x^k, \tag{2.1}$$

where x^k is a given initial point and t a curve parameter. The solution curve to the above system of differential equations is a steepest descent curve in x-space emanating from x^k and tending asymptotically to x^*.

To solve system (2.1) is, generally, not easy owing to the nonlinearity of $g(x)$. However, an approximate solution may be obtained by expanding $g(x)$ in a Taylor series to a first-order approximation, i.e.,

$$g(x) \simeq g(x^k) + H(x^k)(x - x^k). \tag{2.2}$$

The solution curve to the linearized system is then given by

$$x(t) = x^k + [\exp(-tH(x^k)) - I]\, H^{-1}(x^k)\, g(x^k), \tag{2.3}$$

where $\exp(-tH(x))$ is the matrix exponential function of $-tH(x)$.

Let now $\lambda_i(x)$ be the eigenvalues of $H(x)$ and $u_i(x)$ the associated normalized eigenvectors, $i = 1, 2, \ldots, n$. Then

$$[\exp(-tH(x)) - I]\, H^{-1}(x) = \sum_{i=1}^{n} \frac{\exp(-t\lambda_i(x)) - 1}{\lambda_i(x)}\, u_i(x)\, u_i{}^T(x). \tag{2.4}$$

Our basic iteration formula may now be obtained via the following equation:

$$x^{k+1} = x^k + \left[\sum_{i=1}^{n} \frac{\exp(-t(x^k)\, \lambda_i(x^k)) - 1}{\lambda_i(x^k)}\, u_i(x^k)\, u_i{}^T(x^k)\right] g(x^k), \tag{2.5}$$

where $t(x^k)$ is the value of the parameter t minimizing $f(x(t))$ along

$$p(x^k, t) = \left[\sum_{i=1}^{n} \frac{\exp(-t\lambda_i(x^k)) - 1}{\lambda_i(x^k)} u_i(x^k)\, u_i^T(x^k)\right] g(x^k). \tag{2.6}$$

For a complete discussion of the properties of the above curvilinear search path see [1–4].

3. An Iterative Procedure for the Approximation of the Eigensystem of the Hessian Matrix

Let $f(x)$ be a quadratic function of the form

$$f(x) = \tfrac{1}{2}(Qx, x) + (b, x) + c, \tag{3.1}$$

where $Q \in R^{n\times n}$ is a positive definite symmetric matrix. Clearly Q satisfies

$$g(x^{i+1}) - g(x^i) = Q(x^{i+1} - x^i) \tag{3.2}$$

for any two points x^{i+1} and x^i in R^n or, letting $z^i = g(x^{i+1}) - g(x^i)$ and $y^i = x^{i+1} - x^i$,

$$z^i = Qy^i. \tag{3.3}$$

When the above equation is evaluated at $n + 1$ points x^i, $i = 1, 2,..., n + 1$, it yields

$$Z = QY, \tag{3.4}$$

where the columns of Z and Y are z^i and y^i, $i = 1, 2,..., n$, respectively.

Let now λ_i be an eigenvalue of Q and u_i the associated normalized eigenvector. Then

$$Qu_i = \lambda_i u_i\,. \tag{3.5}$$

Taking now the transpose of Eq. (3.4) and premultiplying it by u_i we obtain, since Q is symmetric,

$$Z^T u_i = Y^T Q u_i \tag{3.6}$$

and because of (3.5)

$$Z^T u_i = \lambda_i Y^T u_i\,. \tag{3.7}$$

Hence, finding the eigenvalues and eigenvectors of Q is equivalent to solving the generalized eigenvalue problem (3.7). Note that no matrix inversion is required as Q is not needed explicitly.

Suppose now that $f(x)$ is not quadratic. In this case the difference equation (3.2) does not hold. However, it can be used to provide a reasonable approximation to the Hessian matrix, at least in the direction $y^i = x^{i+1} - x^i$. Note that

Eq. (3.2) has been used in the class of variable metric methods [6–9] to generate approximations to the inverse of the Hessian matrix.

To this end we shall assume that the eigensystem of the Hessian matrix $H(x^k)$, at x^k, satisfies

$$(Z^k)^T u_i^k = \lambda_i^k (Y^k)^T u_i^k, \qquad i = 1, 2, \ldots, n, \tag{3.8}$$

where $(u_i^k, u_j^k) = \delta_{ij}$, since the Hessian matrix is symmetric. At the next point x^{k+1}, obtained through Eq. (2.5), we have

$$(Z^{k+1})^T u_i^{k+1} = \lambda_i^{k+1} (Y^{k+1})^T u_i^{k+1}, \qquad i = 1, 2, \ldots, n. \tag{3.9}$$

The matrices Z^k and Y^k are updated by means of

$$Z^{k+1} = Z^k + (z^k - Z^k e_j) e_j^T$$

and

$$Y^{k+1} = Y^k + (y^k - Y^k e_j) e_j^T,$$

where e_j is a unit vector having unity as the jth element, zeros elsewhere and $j = k + 1$.

Let us now set

$$u_i^{k+1} = u_i^k + \epsilon v_i^k, \qquad i = 1, 2, \ldots, n, \tag{3.10}$$

and

$$\lambda_i^{k+1} = \lambda_i^k + \epsilon \mu_i^k, \qquad i = 1, 2, \ldots, n, \tag{3.11}$$

where ϵ is a perturbation parameter, and v_i^k and μ_i^k are corrections to be determined. Introducing (3.10)–(3.11) into (3.9) we obtain

$$\begin{aligned}(Z^{k+1} - \lambda_i^k Y^{k+1})^T \epsilon v_i^k - \epsilon \mu_i^k (Y^{k+1})^T u_i^k \\ = (-Z^{k+1} + \lambda_i^k Y^{k+1})^T u_i^k + \epsilon^2 \mu_i^k (Y^{k+1})^T v_i^k.\end{aligned} \tag{3.12}$$

If we assume that x^{k+1} is sufficiently close to x^k, then u_i^{k+1} and λ_i^{k+1} are also close to u_i^k and λ_i^k, respectively. Therefore, we can in (3.12) omit higher powers in ϵ, and letting

$$\delta u_i^k = \epsilon v_i^k, \qquad i = 1, 2, \ldots, n, \tag{3.13}$$

and

$$\delta \lambda_i^k = \epsilon \mu_i^k, \qquad i = 1, 2, \ldots, n, \tag{3.14}$$

we have

$$\begin{aligned}(Z^{k+1} - \lambda_i^k Y^{k+1})^T \delta u_i^k - \delta \lambda_i^k (Y^{k+1})^T u_i^k \\ = (-Z^{k+1} + \lambda_i^k Y^{k+1})^T u_i^k, \qquad i = 1, 2, \ldots, n.\end{aligned} \tag{3.15}$$

If we now impose on u_i^{k+1} the normality condition

$$(u_i^{k+1}, u_i^{k+1}) = 1, \qquad i = 1, 2, \ldots, n, \tag{3.16}$$

we obtain, to a first-order approximation,

$$(\delta u_i^k, u_i^k) = 0 \qquad i = 1, 2,..., n. \tag{3.17}$$

Equations (3.15) and (3.17) are written in matrix form as

$$\begin{bmatrix} (u_i^k)^T & \\ (Z^{k+1} - \lambda_i^k Y^{k+1})^T & -(Y^{k+1})^T u_i^k \end{bmatrix} \begin{bmatrix} \delta u_i^k \\ \delta\lambda_i^k \end{bmatrix} = \begin{bmatrix} 0 \\ (-Z^{k+1} + \lambda_i^k Y^{k+1})^T \end{bmatrix} u_i^k. \tag{3.18}$$

Solving the above system we obtain δu_i^k and $\delta\lambda_i^k$. Then u_i^{k+1} and λ_i^{k+1} may be found from

$$u_i^{k+1} = u_i^k + \delta u_i^k \tag{3.19}$$

and

$$\lambda_i^{k+1} = \lambda_i^k + \delta\lambda_i^k. \tag{3.20}$$

As may be seen, system (3.18) must be solved for all eigenvalues and eigenvectors, i.e., for all i, $1 \leqslant i \leqslant n$. Moreover, the resulting u_i^{k+1}, $i = 1, 2,..., n$, will not, in general, form an orthonormal set of vectors in R^n, and therefore an orthonormalization procedure such as the Gram–Schmidt process must be applied to them. System (3.18) is now transformed into a simpler form which also allows us to derive some very interesting results.

Let A be the $(n + 1) \times (n + 1)$ matrix

$$A = \begin{bmatrix} u^T & 0 \\ (Z - \lambda Y)^T & -Y^T u \end{bmatrix} \tag{3.21}$$

(the indices i, k and $k + 1$ are temporarily dropped) and let P be the permutation matrix

$$P = \begin{bmatrix} 0_{1\times n} & 1 \\ I_{n\times n} & 0 \end{bmatrix}, \qquad PP^T = I_{(n+1)\times(n+1)}. \tag{3.22}$$

Then

$$\begin{aligned} B = A^T P &= \begin{bmatrix} u & Z - \lambda Y \\ 0 & -u^T Y \end{bmatrix} \begin{bmatrix} 0_{1\times n} & 1 \\ I_{n\times n} & 0 \end{bmatrix} \\ &= \begin{bmatrix} Z - \lambda Y & u \\ -u^T Y & 0 \end{bmatrix}. \end{aligned} \tag{3.23}$$

The inverse of B may be found using the bordering method for inverting matrices. We obtain

$$B^{-1} = \begin{bmatrix} C^{-1} + sC^{-1} w_q{}^T C^{-1} & -sC^{-1} w \\ -s_q{}^T C^{-1} & s \end{bmatrix}, \tag{3.24}$$

where

$$C = Z - \lambda Y, \tag{3.25}$$

$$w = u, \tag{3.26}$$

$$q^T = -u^T Y, \tag{3.27}$$

and

$$1/s = -q^T C^{-1} w. \tag{3.28}$$

Using the fact that P is an orthogonal matrix, we obtain from (3.23)

$$A^{-1} = (B^T)^{-1} P^T$$

or, because of (3.24),

$$A^{-1} = \begin{bmatrix} -s(C^T)^{-1}q & (C^T)^{-1} + s(C^T)^{-1} qw^T(C^T)^{-1} \\ s & -sw^T(C^T)^{-1} \end{bmatrix}. \tag{3.29}$$

Now using the notation (3.25)–(3.28), system (3.18) is written as

$$A \begin{bmatrix} \delta u \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} 0 \\ -C^T \end{bmatrix} w. \tag{3.30}$$

Hence

$$\begin{bmatrix} \delta u \\ \delta\lambda \end{bmatrix} = A^{-1} \begin{bmatrix} 0 \\ -C^T \end{bmatrix} w = \begin{bmatrix} -w - s(C^T)^{-1}q \\ s \end{bmatrix} \tag{3.31}$$

since $(C^T)^{-1}C^T = I$ and $(w, w) = 1$. The following recursive relations for approximating the eigenvalues and eigenvectors of the Hessian matrix may now be obtained:

$$u_i^{k+1} = s_i^k((Z^{k+1} - \lambda_i^k Y^{k+1})^T)^{-1} (Y^{k+1})^T u_i^k, \tag{3.32}$$

$$\lambda_i^{k+1} = \lambda_i^k + s_i^k, \tag{3.33}$$

where the normalizing factor s_i^k is given by

$$1/s_i^k = (Y^{k+1}(Z^{k+1} - \lambda_i^k Y^{k+1})^{-1} u_i^k, u_i^k) \tag{3.34}$$

for all i, $1 \leqslant i \leqslant n$. If the objective function is of the form (3.1), i.e., quadratic, we have

$$Q = Z^k(Y^k)^{-1} = (Y^k)^T ((Z^k)^T)^{-1}, \qquad k \geqslant n, \tag{3.35}$$

and therefore (3.32) and (3.33) reduce to

$$u_i^{k+1} = s_i^k(Q - \lambda_i^k I)^{-1} u_i^k \tag{3.36}$$

and

$$\lambda_i^{k+1} = \lambda_i^k + s_i^k, \tag{3.37}$$

respectively, $i = 1, 2,..., n$, $k \geqslant n$, where

$$1/s_i^k = ((Q - \lambda_i^k I)^{-1} u_i^k, u_i^k). \tag{3.38}$$

In the following we prove that (3.36)–(3.37) converge to the actual eigenvectors and eigenvalues of Q.

Let μ_i be the eigenvalues and v_i the associated normalized eigenvectors of Q, $i = 1, 2,..., n$. Since Q is symmetric the set $\{v_i \mid i = 1, 2,..., n\}$ constitutes a basis for R^n, and therefore u_i^k can be expanded in terms of the basis vectors as

$$u_i^k = (1 - c_i^k) v_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} c_j^k v_j , \tag{3.39}$$

where

$$(1 - c_i^k)^2 + \sum_{\substack{j=1 \\ j \neq i}}^{n} (c_j^k)^2 = 1, \tag{3.40}$$

since u_i^k is a unit vector. Suppose now that λ_i^k is closer to μ_i than to any of the other eigenvalues of Q, and consider the expansion of s_i^k in (3.38) into the eigenspaces of Q. We have

$$\frac{1}{s_i^k} = \frac{(1 - c_i^k)^2}{\mu_i - \lambda_i^k} + \sum_{\substack{j+1 \\ j \neq 1}}^{n} \frac{(c_j^k)^2}{\mu_j - \lambda_i^k} \tag{3.41}$$

$$= \frac{1}{\mu_i - \lambda_i^k} \left[(1 - c_i^k)^2 + \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\mu_i - \lambda_i^k}{\mu_j - \lambda_i^k} (c_j^k)^2 \right] \tag{3.42}$$

and, because of (3.40),

$$\frac{1}{s_i^k} = \frac{1}{\mu_i - \lambda_i^k} (1 - \gamma_i^k), \tag{3.43}$$

where

$$\gamma_i^k = \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(1 - \frac{\mu_i - \lambda_i^k}{\mu_j - \lambda_i^k} \right) (c_j^k)^2. \tag{3.44}$$

Introducing (3.44) into (3.37) we obtain

$$(1 - \gamma_i^k) \lambda_i^{k+1} = (1 - \gamma_i^k) \lambda_i^k + \mu_i - \lambda_i^k, \tag{3.45}$$

from which

$$| \lambda_i^{k+1} - \mu_i | = | \gamma_i^k/(1 - \gamma_i^k) | \, | \lambda_i^k - \mu_i | . \tag{3.46}$$

Note that $\gamma_i^k > 0$, since $\rho_{j,i}^k = |(\mu_i - \lambda_i^k)/(\mu_j - \lambda_i^k)| < 1, j = 1, 2,..., n, j \neq i$, and that $\lim_{k\to+\infty} | \lambda^{k+1} - \mu_i | = 0$ if $r_i^k = | \gamma_i^k/(1 - \gamma_i^k) | < 1$.

Let us now assume that u_i^k is ϵ-close to v_i, i.e.,

$$c_i^k = \epsilon_i^k, \; | \epsilon_i^k | \leqslant \epsilon, \qquad i = 1, 2,..., n. \tag{3.47}$$

Then, clearly, $r_i^k < 1$, and it follows from (3.46) that λ_i^{k+1} is a better approximation to μ_i than λ_i^k was. Moreover, λ_i^k converges at least linearly to μ_i, the rate of convergence depending upon r_i^k, i.e., upon both ϵ_i^k and $\rho_{j,i}^k$. Let now $\rho_{l,i}^k = |(\mu_i - \lambda_i^k)/(\mu_l - \lambda_i^k)| \ll 1$, where $| \mu_l - \lambda_i^k | = \min_{j\neq i} | \mu_j - \lambda_i^k |$. Note that $\rho_{l,i}^k \ll 1, i = 1, 2,..., n$, if the eigenvalues of Q are far apart. In this case (3.44) reduces to

$$\gamma_i^k = \sum_{\substack{j=1 \\ j\neq i}}^{n} (\epsilon_j^k)^2 \ll 1 \tag{3.48}$$

and therefore

$$r_i^k = \frac{\sum_{j=1, j\neq i}^{n} (\epsilon_j^k)^2}{1 - \sum_{j=1, j\neq i}^{n} (\epsilon_j^k)^2} \simeq \sum_{\substack{j=1 \\ j\neq i}}^{n} (\epsilon_j^k)^2 \ll 1. \tag{3.49}$$

Hence it follows from (3.46) and (3.49) that in the neighborhood of each eigenvector the convergence to μ_i is very fast. Moreover, if $\epsilon_j^k = 0, j = 1, 2,..., n, j \neq 1$, i.e., if $u_i^k = v_i$, then $\lambda_i^{k+1} = \mu_i$, a result that may also be directly obtained from (3.37) by letting $u_i^k = v_i$.

Let us now examine the next approximation to v_i given by Eq. (3.36). From (3.37) we have $s_i^k = \lambda_i^{k+1} - \lambda_i^k$, and therefore (3.36) is written as

$$u_i^{k+1} = (\lambda_i^{k+1} - \lambda_i^k) \left(\frac{1 - \epsilon_i^k}{\mu_i - \lambda_i^k} v_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\epsilon_j^k}{\mu_j - \lambda_i^k} v_j \right). \tag{3.50}$$

Now, from (3.45) we obtain

$$\lambda_i^{k+1} - \lambda_i^k = (\mu_i - \lambda_i^k)/(1 - \gamma_i^k). \tag{3.51}$$

Hence

$$u_i^{k+1} = \frac{1}{1 - \gamma_i^k} \left[(1 - \epsilon_i^k) v_i + \sum_{\substack{j=1 \\ j\neq 1}}^{n} \frac{\mu_i - \lambda_i^k}{\mu_j - \lambda_i^k} \epsilon_j^k v_j \right] \tag{3.52}$$

and, clearly, u_i^{k+1} is richer in v_i than u_i^k was. From (3.45) we also have

$$\mu_i - \lambda_i^k = -\frac{\gamma_i^k}{1-\gamma_i^k}\frac{(\mu_i-\lambda_i^k)^2}{\mu_i-\lambda_i^{k+1}} \tag{3.53}$$

and thus from (3.52), using (3.44) as well,

$$u_i^{k+1} = \frac{1}{1-\gamma_i^k}\left[(1-\epsilon_i^k)v_i - \frac{(\mu_i-\lambda_i^k)^2}{(1-\gamma_i^k)(\mu_i-\lambda_i^{k+1})} \times \sum_{\substack{j=1\\ j\neq i}}^{n}\left[\frac{1}{\mu_j-\lambda_i^k}\left(\sum_{\substack{j=1\\ j\neq i}}^{n}\left(1-\frac{\mu_i-\lambda_i^k}{\mu_j-\lambda_i^k}\right)(\epsilon_j^k)^2\right)\epsilon_j^k v_j\right]\right]. \tag{3.54}$$

As may be seen, the process is cubically convergent, since all coefficients other than that of v_i are cubic in ϵ_j^k, whereas the rate of cubic convergence again depends upon $\rho_{l,i}^k$. Note finally that if $\lambda_i^k = \mu_i$, then u_i^{k+1} is a vector in the direction of v_i.

So far, the convergence properties of our iterative procedure have been established under the assumption that u_i is sufficiently close to v_i, for all i, $1 \leqslant i \leqslant n$. However, as our numerical results indicated, (3.36)–(3.37) never failed to converge. The reason for this may be that for all our test functions the eigenvalues of the actual Hessian matrix are well separated. In this case we obtain $\gamma_i^k = \sum_{j=1, j\neq i}^{n}(\epsilon_j^k)^2$, and for r_i^k to be less than unity we must have $\sum_{j=1,j\neq i}^{n}(\epsilon_j^k)^2 < \frac{1}{2}$ or $(u_i^k, v_i) > 2^{1/2}/2$; i.e., the angle between u_i^k and v_i must be smaller than $\pi/4$, a condition which is not difficult to satisfy. Note also that because of the Gram–Schmidt orthonormalization procedure, (3.36) is actually replaced by the following equations:

$$\begin{aligned}
&\text{(a)} \quad w_i^{k+1} = s_i^k(Q-\lambda_i^k I)^{-1}u_i^k, \qquad i = 1, 2, \ldots, n,\\
&\text{(b)} \quad \bar{u}_1^{k+1} = w_1^{k+1},\\
&\text{(c)} \quad \bar{u}_i^{k+1} = w_i^{k+1} - \sum_{j=1}^{i-1}(w_i^{k+1}, \bar{u}_j^{k+1})\bar{u}_j^{k+1}, \qquad i = 2, 3, \ldots, n,\\
&\text{(d)} \quad u_i^{k+1} = \bar{u}_i^{k+1}/\|\bar{u}_i^{k+1}\|, \qquad i = 1, 2, \ldots, n.
\end{aligned} \tag{3.55}$$

As may be easily verified, if the approximated eigenvectors are ϵ-close to the actual ones, then the inner products in (3.55c) are of order $O(\epsilon^6)$, and therefore the recursive equations (3.55a)–(3.55d) exhibit the same properties as (3.36).

4. The Algorithm

An algorithm using iteration formula (2.5), with the eigensystem of the Hessian matrix being approximated as in the previous section, has been written and

tested extensively.[1] The eigenvalues and eigenvectors of the Hessian matrix are approximated by solving system (3.18) and then using (3.19)–(3.20), instead of directly updating according to (3.32)–(3.33), so as to avoid the undesirable procedure of matrix inversion. Then the new approximations to the eigenvalues are placed in ascending order, and the resulting set of eigenvectors after rearrangement is orthonormalized using the Gram–Schmidt process. The initial approximations to the eigenvectors are set equal to $u_i^0 = e_i$, $i = 1, 2, \ldots, n$, where e_i is a unit vector with unity in the ith position and zeros elsewhere, whereas the λ_i^0, $i = 1, 2, \ldots, n$, are distributed over the interval $[1, n]$ according to $\lambda_i^0 = i/(n + 1 - i)$. The algorithm is reset when system (3.18) is singular or causes certain eigenvalues to fall outside some lower and upper bounds, or when the resulting set of eigenvectors is linearly dependent.

In order to determine the next approximation x^{k+1} to the minimizer x^*, we can minimize the objective function along our curvilinear search path (2.6) emanating from x^k. This is, however, a most inefficient way of generating successive approximations to x^*, as it involves an excessive number of function evaluations. In some of the minimization algorithms developed so far an attempt was made to overcome this disadvantage by making use of a cubic interpolation in order to predict x^{k+1}, but if this is done the gradient vector has to be computed at all intermediate points. In our algorithm the following procedure is used, called "quadratic interpolation." We first, in (2.6), replace the λ_i^k, $i = 1, 2, \ldots, n$, by their absolute values, thus forcing the Hessian matrix to be positive definite. This also allows us to let $t \to +\infty$, yielding

$$x^{k+1} = x^k - H^{-1}(x^k)\, g(x^k) = x^k - \left[\sum_{i=1}^{n} \frac{u_i^k (u_i^k)^T}{\lambda_i^k}\right] g(x^k).$$

Note that this is the point obtained when Newton's method is used with a step-size of unity. If $f(x^k - H^{-1}(x^k)\, g(x^k)) < f(x^k)$, we proceed to the next iteration; otherwise a quadratic interpolation is used repeatedly until a decrease in the function value is obtained.

We do this by making the change of variable $t = \tau/(1 - \tau)$ and assuming that $f(x(\tau))$ varies quadratically between $\tau = 0$ and $\tau = \tau_s^k$ whenever $f(x(\tau_s^k)) > f(x(0))$, where s denotes the number of steps in the unidimensional search. The quadratic approximation to the objective function along our curvilinear search path (2.6) is obtained by letting $f(x(0)) = f(x^k)$, $f(x(\tau_s^k)) = f(x_s^k)$, and $df(x(0))/d\tau = -\| g(x^k)\|^2$. As may be proved, $f(x(\tau))$ has a minimum at $\tau_{s+1}^k = \tau_s^k(1 - (\gamma^k + \delta_s^k)/(2\gamma^k + \delta_s^k))$, where $\gamma^k = \| g(x^k)\|^2 > 0$ and $\delta_s^k = (2/\tau_s^k)\,(f(x_s^k) - f(x^k)) > 0$. To abviate the possibility of the interpolated minimum falling quite close to the point from where the interpolation was

[1] For a tape, as well as a listing of the program, apply to the author at the following address: National Research Institute for Mathematical Sciences, CSIR, P.O. Box 395, Pretoria 0001, South Africa.

started, we set $\tau_{s+1}^k = \max\{0.1\ \tau_s^k, \tau_{s+1}^k\}$. We then set $x_{s+1}^k = x^k + p(x^k, t_{s+1}^k)$ and test whether the condition $(f(x_{s+1}^k) - f(x^k))/(x_{s+1}^k - x^k, g(x^k)) \geqslant \epsilon$ is satisfied, where ϵ is a small positive number. If this is the case we set $x^{k+1} = x_{s+1}^k$ and $t(x^k) = t_{s+1}^k$, as a sufficient function decrease has been achieved; otherwise we set $s = s + 1$ and repeat the quadratic interpolation. A suitable initial approximation to the step-size is given by $\tau_0^k = \min\{1, \tau^k\}$ where τ^k is the root of

$$2f(x^k) = \sum_{i=1}^{n} \frac{1 - \exp(-t\lambda_i^k)}{\lambda_i^k} (g(x^k), u_i^k)^2, \qquad t = \tau/(1 - \tau),$$

unless $f(x^k) \leqslant 0$, in which case we take $\tau_0^k = 1$, i.e., $t \to \infty$.

The fact that some approximations to the eigenvalues and eigenvectors of the Hessian matrix are available may now be used to accelerate the rate ofconvergence and to get past stationary or almost-stationary points which are not relative minima. We recall that at such a stationary point the gradient vector is zero, but the Hessian matrix is indefinite. Indeed, let $\| g(x^k)\|$ be almost zero at a certain point x^k of our minimization process and let $\lambda_i^k < 0$ for some i, $1 \leqslant i \leqslant n$. Then assuming that our approximation scheme of the previous section approximates the eigensystem of the actual Hessian matrix reasonably well, the region around x^k is an almost-stationary point which is not a relative minimum (case of a saddle point). As may now be seen $(\exp(-t\lambda_i^k) - 1)/\lambda_i^k \to -\infty$ as $t \to +\infty$, and therefore the dominant term in the expansion (2.6) is that corresponding to u_i^k. This suggests that u_i^k should be used as the search direction during the next iteration, which is a result that could, however, have been expected, since the best downhill direction in the case of a saddle point is the eigenvector corresponding to the most negative eigenvalue. As numerical results have indicated, the use of one of the eigenvectors is also suggested when $\lambda_i^k = 0$, $1 \leqslant i \leqslant n$ (x^k is a point of singularity of the Hessian matrix) or, more generally, when $|\lambda_{\min}^k/\lambda_{\max}^k| \ll 1$ (x^k lies on a valley in the direction of $u_{\min}^k$). Note that if $|\lambda_{\min}^k/\lambda_{\max}^k| \ll 1$, then (2.6) reduces to $x(t) \simeq (\exp(-t\lambda_{\min}^k) - 1/\lambda_{\min}^k)(u_{\min}^k, g(x^k))u_{\min}^k$, which is a result suggesting that $u_{\min}^k$ should be used as the search direction, provided that $(u_{\min}^k, g(x^k)) \neq 0$. However, since our iteration formula (2.6) is well defined even when the Hessian matirx is singular or non-positive definite, we consider the possibility of using one of the eigenvectors at x^k only when $|(f(x^{k-1}) - f(x^k))/f(x^{k-1})| < \epsilon$, where ϵ is a small positive number.

The following algorithm may now be stated.

STEP 0. Select an $x^0 \in R^n$ such that the level set $L_0 = \{x \mid f(x) < f(x^0), x \in R^n\}$ is compact; set the tolerances eps 1, eps 2, eps3, eps 4, eps 5, eps 6, eps 7, eps 8, l, L, N, and η.

STEP 1. Set $k = 0, j = 1, Y^0 = \text{diag}(n + 1 - i), Z^0 = \text{diag}(i), i = 1, 2, \ldots, n$

Comment. As may be seen, $\lambda_i^0 = i/n + 1 - i$ and $u_i^0 = e_i$, $i = 1, 2,..., n$.

STEP 2. Set $P(x^0, t) = -tg(x^0)$; go to Step 4.

STEP 3. Solve the system

$$\begin{bmatrix} (u_i^k)^T & 0 \\ (Z^k - \lambda_i^k Y^k)^T & -(Y^k)^T u_i^k \end{bmatrix} \begin{bmatrix} \delta u_i^k \\ \delta\lambda_i^k \end{bmatrix} = \begin{bmatrix} 0 \\ (-Z^k + \lambda_i^k Y^k)^T \end{bmatrix} u_i^k, \qquad i = 1, 2,..., n;$$

if the above system is singular, then set $x^0 = x^{k+1}$ and go to Step 1; otherwise update the eigensystem of the Hessian matrix by means of $w_i^{k+1} = u_i^k + \delta u_i^k$ and $\bar{\lambda}_i^{k+1} = \lambda_i^k + \delta\lambda_i^k$, $i = 1, 2,..., n$; if $|\bar{\lambda}_i^{k+1}| < l$ or $|\bar{\lambda}_i^{k+1}| > L$, $i = 1, 2,..., n$ set $x^0 = x^{k+1}$ and go to Step 1; otherwise rearrange the eigenvalues in increasing order, i.e., $\lambda_i^{k+1} < \lambda_j^{k+1}$, $i < j$, $i, j = 1, 2,..., n$, and orthogonalize the set w_i^{k+1}, $i = 1, 2,..., n$, using the Gram–Schmidt process; let $\bar{u}_i^{k+1}$, $i = 1, 2,..., n$, be the resulting set of vectors, i.e., $\bar{u}_1^{k+1} = w_1^{k+1}$, $\bar{u}_i^{k+1} = w_i^{k+1} - \sum_{j=1}^{i-1} (w_i^{k+1}, \bar{u}_j^{k+1})\bar{u}_j^{k+1}$, $i = 2, 3,..., n$; if the set $\{w_i^{k+1}, i = 1, 2,..., n\}$ is linearly dependent, set $x^0 = x^{k+1}$ and go to Step; otherwise set $u_i^{k+1} = \bar{u}_i^{k+1}/\| \bar{u}_i^{k+1} \|$, $i = 1, 2,..., n$, and $k = k + 1$; go to Step 7.

Comment. In order to solve our systems of linear equations we use Gaussian elimination with partial pivoting. If the diagonal elements of the resulting upper triangular system are all greater in absolute value than eps $2 = 10^{-16}$, we say that the particular system is nonsingular. Before orthonormalizing the set w_i^{k+1}, $i = 1, 2,..., n$, we rearrange the eigenvalues in increasing order, so that the basis vector for the Gram–Schmidt process will be the one corresponding to the smallest eigenvalue. When testing for linear independence, the norm of $\bar{u}_i^{k+1}$, $i = 1, 2,..., n$, is compared with eps $3 = 10^{-24}$, and if $\| \bar{u}_i^{k+1} \| >$ eps 3, $i = 1, 2,..., n$, we say that the set is linearly independent. The upper and lower bounds for the eigenvalues were set equal to $L = 10^6$ and $l = 10^{-16}$, respectively.

STEP 4. If $f(x^k) \leqslant 0$, go to Step 5; otherwise use the bisection method to solve the equation $2f(x^k) = -(p(x^k, t), g(x^k))$, $t = \tau/(1 - \tau)$, and let τ^k be its root; set $s = 0$, $\tau_0^k = \min\{1, \tau^k\}$ and go to Step 5.

Comment. Before applying the bisection method we test whether a root exists in the t-interval $[0, +\infty)$ or equivalently in the τ-interval $[0, 1]$. We do this by testing whether $2f(x^k) < \sum_{i=1}^{n} (g(x^k), u_i^k)^2/\lambda_i^k$, and if this is the case we set $\tau_0^k = 1$. Note that when $p(x^k, t) = -tg(x^k)$, then $\lambda_i^k = 1$ and $u_i^k = e_i$, $i = 1, 2,..., n$, and no change of variable is needed.

STEP 5. (Quadratic interpolation). (i) Set $x_s^k = x^k + p(x^k, t_s^k)$, $t_s^k = \tau_s^k/(1 - \tau_s^k)$.

(ii) Compute $f(x_s^k)$; if $(f(x_s^k) - f(x^k))/(x_s^k - x^k, g(x^k)) \geqslant 10^{-4}$, go to step (iv); otherwise set $\tau_{s+1}^k = \max\{0.1\ \tau_s^k, \bar{\tau}_{s+1}^k\}$, where

$$\bar{\tau}_{s+1}^k = \tau_s^k \left(1 - \frac{\gamma^k + \delta_s^k}{2\gamma^k - \delta_s^k}\right), \qquad \gamma^k = \| g(x^k)\|^2,$$

$$\delta_s^k = \frac{2}{\tau_s^k}(f(x_s^k) - f(x^k)),$$

and go to step (iii);

(iii) Set $s = s + 1$ and go to step (i).

(iv) Set $x^{k+1} = x_s^k$, $t(x^k) = t_s^k = \tau_s^k/(1 - \tau_s^k)$, and go to Step 6.

STEP 6. Compute $f(x^{k+1})$ and $g(x^{k+1})$; if $\| g(x^{k+1})\| \leqslant \text{eps } 1 = 10^{-6}$, stop; otherwise set $Z^{k+1} = Z^k + (z^k - Z^k e_j)e_j^T$, $Y^{k+1} = Y^k + (y^k - Y^k e_j)e_j^T$, where $z^k = g(x^{k+1}) - g(x^k)$ and $y^k = z^{k+1} - x^k$; if $j = n$, then set $j = 1$; otherwise set $j = 1$; otherwise set $j = j + 1$ and go to Step 3.

STEP 7. *Comment.* In this step a decision is to be made as to whether one of the eigenvectors should be used as the search direction during the next iteration. We first determine whether x^k is a saddle point or not.

If $\| g(x^k)\| < \text{eps } 4 = 10^{-4}$ and $\lambda_1^k < 0$, set $d^k = u_1^k$ and go to Step 8. *Comment.* We now examine the possibility of a vanishing Hessian. If $|\lambda_i| < \text{eps } 5 = 5 \times 10^{-2}$, $i = 1, 2, \ldots, n$, go to Step 9. *Comment.* The possible use of one of the eigenvectors is considered only if the relative function decrease is less than eps $6 = 5 \times 10^{-2}$. If $|(f(x^{k-1}) - f(x^k))/f(x^{k-1})| > \text{eps } 6$, go to Step 9. *Comment.* One of the eigenvectors will now be used if x^k is a point of singularity of the Hessian matrix or a point that lies on a steep valley, unless this was the case during the previous $N = 3$ iterations. If $|\lambda_{\min}^k/\lambda_{\max}^k| < \text{eps } 7 = 10^{-4}$, use $u_{\min}^k$, i.e., $d^k = u_{\min}^k$, unless $|(u_{\min}^k, g(x^k))| < \text{eps } 8 \times 10^{-3}$ in which case use the eigenvector corresponding to the next in absolutely increasing order eigenvalue. *Comment.* Finally the existence of a flat saddle point is to be examined. If $\| g(x^k)\| < \eta = 1$ and $\lambda_1^k < 0$, set $d^k = u_1^k$; otherwise go to Step 9; go to Step 8.

STEP 8. The next iterate x^{k+1} is obtained along the direction d^k, being one of the eigenvectors, by repeatedly reducing the step-size by means of $t_s = 1/2^s$, $t_0 = 1$, until a smaller function value is obtained; go to Step 6.

STEP 9. Set the curvilinear search path

$$p(x^k, t) = \left[\sum_{i=1}^{n} \frac{\exp(-t\,|\lambda_i^k|) - 1}{|\lambda_i^k|}\, u_i^k u_i^k\right] g(x^k)$$

and go to Step 4.

The convergence properties of this algorithm, when the quadratic interpolation is replaced by a unidimensional minimization, have been established by a theorem presented in [3].

The algorithm described above has been programmed and executed on a number of test functions. The numerical results indicate that:

(i) the new algorithm is very fast, stable, and capable of overshooting stationary or almost-stationary points and points of singularity of the Hessian matrix, and moves rapidly toward the minimum along a narrow valley; and

(ii) the eigensystem of the Hessian matrix is well approximated, especially in the vicinity of the minimum, whereas for quadratic functions the actual eigenvalues and eigenvectors are obtained after a few iterations.

Our algorithm has also been compared with the standard IBM subroutine of Fletcher's method [6], and the results indicate that the new algorithm is preferable: Not only was the total number of evaluations smaller, but it was possible to find the minimum in cases where Fletcher's algorithm failed.

The following is an extract from our numerical results; these illustrate the typical behavior of the algorithm. Complete numerical results involving experiments on 14 well-known functions may be obtained from the author [11].

In Tables I and II the total number of evaluations is equal to the number of function evaluations plus n times the number of gradient evaluations.

TABLE I

New Algorithm[a]

Function	Starting point	Iterations	Evaluations
Parabolic valley	$(-1.2, 1)$	35	138
	$(3, 3)$	26	94
	$(8, 8)$	57	219
Cubic valley	$(-1.2, -1)$	39	159
	$(3, 3)$	70	296
	$(8, 8)$[b]	98	399
Cliff function[b]	$(0, -1)$	36	153
	$(0, 0)$	17	97
Quartic with singular Hessian	$(3, -1, 0, 1)$	47	260
Botsaris function	$(-1.2, -1, -1.2, \ldots, -1)$	95	1195

[a] Near the minimum the eigensystem of the Hessian matrix were very well approximated by our iterative process.

[b] Fletcher's algorithm failed to converge.

TABLE II

Fletcher's Algorithm

Function	Starting point	Iterations	Evaluations
Parabolic valley	$(-1.2, 1)$	37	141
	$(3, 3)$	48	180
	$(8, 8)$	96	381
Cubic valley	$(-1.2, -1)$	48	174
	$(3, 3)$	133	534
	$(8, 8)$[a]	377	1467
Cliff[a] function	—	—	—
	—	—	—
Quartic with singular Hessian	$(3, -1, 0, 1)$	52	275
Botsaris function	$(-1.2, -1, -1.2, \ldots, -1)$	233	2948

[a] Fletcher's algorithm failed to converge.

Parabolic valley.

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,\ x^* = (1, 1)^T,\ f(x^*) = 0.$$

Cubic valley.

$$f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2,\ x^* = (1, 1)^T,\ f(x^*) = 0.$$

Cliff function.

$$f(x) = \left(\frac{x_1 - 3}{100}\right)^2 - (x_1 - x_2) + e^{20(x_1 - x_2)}, \qquad x^* = (3, 3.314978)^T,$$

$$f(x^*) = 0.1997866.$$

Quartic with singular Hessian.

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4,$$
$$x^* = (0, 0, 0, 0)^T,\ f(x^*) = 0.$$

Botsaris function.

$$f(x) = 100\left[x_n - \left(\frac{1}{n-1}\sum_{i=1}^{n-1} x_i\right)^2\right]^8 + \sum_{i=1}^{n-1}(1 - x_i)^8, \qquad x^* = (1, 1, \ldots, 1)^T,$$

$$f(x^*) = 0.$$

This function was minimized for $n = 10$.

REFERENCES

1. C. A. BOTSARIS AND D. H. JACOBSON, A Newton-type curvilinear search method for optimization, *J. Math. Anal. Appl.* **54** (1976), 217–229.
2. C. A. BOTSARIS AND D. H. JACOBSON, Minimization of a nonlinear function in many variables: A differential equation approach, paper presented at a "Symposium on Differential Equations held at the Council for Scientific and Industrial Research, Pretoria, April 1975."
3. C. A. BOTSARIS, Differential gradient methods, *J. Math. Anal. Appl.* **63** (1978), 177–198.
4. C. A. BOTSARIS, "Differential Descent Methods for Function Minimization," Doctoral thesis, University of the Witwatersrand, Johannesburg, South Africa, August 1975.
5. J. H. WILKINSON, "The Algebraic Eigenvalue Problem," Oxford Univ. Press, London/New York, 1965.
6. R. FLETCHER, A new approach to variable metric algorithms, *Comput. J.* **13** (1970), 317.
7. W. C. DAVIDON, Variable metric methods for minimization, A.E.C. Research and Development Report No. ANL-5990 (Rev.) 1959.
8. C. G. BROYDEN, Quasi-Newton methods and their application to function minimization, *Math. Comp.* **21** (1967), 368.
9. R. FLETCHER AND M. J. D. POWELL, A rapidly convergent descent method for minimization, *Comput. J.* **6** (1963), 163.
10. D. M. HIMMELBLAU, "Applied Nonlinear Programming," McGraw–Hill, New York, 1972.
11. C. A. BOTSARIS, A comparison study of a new algorithm for function minimization, CSIR Special Report WISK 214, July 1976.